**Madhu**

T: 862 216 0604

E: madhupavan7@gmail.com

**Technical Skills**

JavaScript, Backbone, AngularJS, Node, HTML5, CSS3, SASS, jQuery, JSON, Backbone.js, Node.js, Bootstrap, Dojo, XML, XSLT, Ajax, CSS3, Java, Perl, Python, PHP, C, C++, PL/SQL, MongoDB, Transact-SQL, Oracle, MySQL, SQLite, Sybase, Bash, Catalyst, Apache, Tomcat, IIS, JBoss, Web logic, WebSphere Application Server, WebSphere MQ, Apache Ant, Maven, GIT, CVS, SVN, UNIX (Linux, Solaris, AIX), Windows, OpenSSH, Kerberos, Bugzilla, Awk, Sed, Spring MVC, JSP, Servlets, EJB, Seam, Hibernate, JDBC, JUnit, Eclipse, NetBeans, JBDS, Catalyst, Mojolicious, Moose.

**Summary**

* Over 7 years of Software Development experience
* Over 5 years of UI development experience
* Worked on Responsive Web Design
* Worked with HTML5, CSS3, jQuery, Backbone, AngularJS, Node, Dojo
* Experience in developing Online Banking and NACHA/ACH applications
* Worked with MVC frameworks such as Backbone, AngularJS
* Developed applications in Java /JEE using Spring, Hibernate, JSP, Servlets
* Developed software using Object Oriented programming techniques
* Worked in Online Banking, NACHA / ACH, Real Time Banking and Wire Transfers domains
* Performed Web Development using HTML5, CSS3 and JavaScript libraries, jQuery, Bootstrap.
* Worked with Oracle, MySql, SQLite, Sybase database systems, performed database tuning and administration
* Worked with Apache, Tomcat, JBoss, IIS servers
* Worked on UNIX, Linux and Windows platforms
* Performed UNIX / Linux system administration
* Worked with OpenSSH, GPG, PGP, Kerberos, created and maintained SSL Certs
* Worked with different Revision Control Systems and performed code deployment

**Experience**

**CGI Federal** Sep 2013 - Present

**Herndon, VA**

UI Developer

**Obamacare**

Enhancements to the Healthcare.gov web site to implement Obamacare.

*Accomplishments*

* Designed and developed wire frames for the UI specifications
* Developed UI screens using HTML5 and CSS3
* Created a responsive UI using Bootstrap and jQuery
* Used Ajax, jQuery to develop a REST framework
* Developed Backbone Models, Views and Collections employing Underscore templating
* Developed widgets using jQuery UI and developed custom jQuery plugins
* Developed custom rich text editor using HTML5, Bootstrap and jQuery
* Worked with JBoss and Tomcat servers Developed RESTful web services in Java
* Used Apache velocity templating engine

**First Data – FundsXpress** Jan 2009 – Aug 2013

**Austin, TX**

Senior Perl Developer

**Banner Store**

A feature rich Seam and JEE based web application which provides banking administrators the ability to control the background image of the online banking outer frame and to arrange for the rotation of ads within campaigns.

*Accomplishments*

* Used JBoss and Seam application framework
* Developed the view using Spring MVC
* Used hibernate for ORM and created the mapping configuration file
* Designed and developed database schema
* Developed JavaScript framework governing the banners
* Used jQuery and Bootstrap for creating effects on banners and ads.
* Used JUnit to perform unit and regression tests
* Built the project using Apache Ant
* Implemented user authentication using single sign-on

**Banner Server**

A light-weight JDBC and Servlet based web application which serves the outer frame HTML and its background image and ads to online banking customers.

*Accomplishments*

* Used JSP for developing the web interface.
* Used JDBC for database access
* Designed and developed database schema
* Developed and deployed servlet on JBoss app server
* Used Apache Ant for building
* Implemented abstract template classes for JUnit

**UI Refresh**

Revamp the online banking web site to be modern and more user friendly. The new web interface will make the pages look more snappy and will improve the experience of the online banking customer by including dialogue boxes, sliders, fancy boxes, widgets, improved dropdowns, buttons, accordions. A new interactive snapshot page will be added, which will act as an aggregator for all the accounts of the customer. The user will be able to customize the look and feel of the web interface from the snapshot config tool.

*Accomplishments*

* Implemented the UI changes using HTML5 and CSS3 and jQuery libraries, jQuery UI and plugins
* Developed JavaScript API using Backbone and Underscore
* Used jQuery to build a dashboard / snapshot page for online banking application
* Developed modals and interactive boxes with jQuery plugins
* Developed action menus using HTML5 and jQuery
* Implemented field validations using HTML5
* Improved the page JavaScript response by using frame to carry over JavaScript to all pages
* Developed custom jQuery plugins for select boxes
* Added dynamic data fetching capabilities to the web site using Ajax
* Used jQuery plugins like SelectBox, FancyBox, DatePicker, CountDown, ToolBar
* Designed and developed JavaScript API
* Developed framework to combine JavaScript and jQuery code into one single package
* Minified the JavaScript library file using minify framework
* Developed JavaScript syntax checker framework using Jlint
* Implemented the MVC architecture using Perl Catalyst
* Developed a session timeout feature for the online banking sessions
* Implemented real time account data features using Ajax and jQuery
* Leveraged HTML5 features to implement field validation and secure browsing
* Improved the response of pages by using jQuery plugins
* Implemented Perl API to process JavaScript using JSON module
* Used GIT for revision control
* Developed regression test framework using Selenium WebDriver in Perl

**Nacha**

Enhancements to the existing NACHA based ACH capabilities of online banking application to support PPD, CCD, CTX, CIE, ARC, POP, BOC, RCK, TEL and WEB standard entry codes. The existing ACH backend processing framework was updated to reconcile the Fed NACHA changes. This will enable online banking customers to create ACH batches, import ACH batches with the new SEC codes, create sub users and set limit restrictions on transactions. With this project, financial institutions will be able to generate/process multiple ACH files at different cutoff times of the day in comparison to the single cutoff time of the system. This feature enables financial institutions to implement aggressive risk based ACH transaction for their commercial customers.

*Accomplishments*

* Drafted the functional design document according to SRS to comply with the NACHA rules and regulations
* Created the use cases and class relationship diagram in UML
* Developed real time processing framework using EJB on NetBeans
* Developed and deployed servlet on JBoss app server for real time transactions
* Developed Object Oriented Perl API for ACH batch processing.
* Implemented new scheduler for generating ACH Fed Onus and Tran files
* Developed a Job processing API in Perl for processing ACH batches
* Automated Jobs using shell scripts and reduce the load on servers during peak batch processing
* Developed an Object Oriented Perl framework for database interaction using Perl DBI
* Designed and developed Status logging interface
* Designed and developed user interface using Perl CGI, HTML, JavaScript, jQuery and MIC, an in-house Perl Template toolkit
* Improved the front end response time in handling multiple batch form pages
* Implemented multi-factor authentication on transaction oriented pages
* Enhanced the capabilities of MIC (templating toolkit) based on Perl Catalyst
* Developed test framework using Perl Test::Harness.
* Designed and implemented database schemas
* Created stored procedures, functions and cursors
* Designed ACH Batch validation framework for the financial institutions to check the integrity and conformity of the ACH batches.
* Maintained and handled the project branches on CVS
* Used Make build the project and maintained the Makefiles
* Employed GPG utility to encrypt the ACH files and reports for Fed.

**Security Token**

Integrate VeriSign hard/soft token into the online banking application to provide portable token multi-factor authentication. This enhancement would provide an additional security layer to online banking customers when accessing their accounts. This feature is also integrated with financial institution’s back-end banking applications. The API should also be capable of handling soft token.

*Accomplishments*

* Performed functional analysis and developed functional design document as per the software specifications
* Developed use case diagrams and proof of concept
* Developed hardware security token simulation tool using VeriSign’s test pilot development API
* Created WSDL and drafted XSD for the web services
* Customized SOAP::Lite CPAN module to create a web services API in Perl
* Integrated security tokens into the existing multi-factor authentication methods to work transparently
* Developed UI using Perl CGI, HTML, JavaScript, CSS, jQuery and Ajax
* Implemented database changes and performed database optimization
* Handled the application deployment from pilot to production environment
* Assisted Quality Assurance team in setting up test environment with VeriSign API
* Developed a custom token simulation tool to mimic the SOAP web services in Perl
* Designed and developed a front end token management system using HTML, JavaScript and Ajax for bank-end banking applications
* Used SQLite for storing token information to reduce the overhead on existing database system
* Implemented SSO interface for customers to manage tokens on VeriSign from the banking application seamlessly

**Wire Transfers**

Enhancement of the existing wire transfer application to support wire acknowledgements, incoming wire transfers and provide financial institutions a configuration option which allows them to interface with Fiserv for wire transfer processing. This enhancement will enable financial institutions to choose Fiserv as processing core for both incoming and outgoing wires.

*Accomplishments*

* Developed a dedicated framework to set up financial institutions to use SFTP for accessing incoming and outgoing Fed formatted wire transfer files
* Developed API for interfacing with third party wire processing providers like Fiserv
* Updated the existing wire application to conform to the new Fed rules
* Generated new reports using Perl and PL/SQL procedures
* Developed Single Sign On framework to interface with Fiserv
* Developed front-end tool to monitor and manage wire transfers with different cores
* Created and maintained SSL Certs
* Developed unit and regression test scripts
* Provided 24/7 production support

**Heartbeat Digital** May 2008 - Dec 2009

**New York City, NY**

Perl / UI Developer

**Pub-Med / NCBI**

Design a web based user interface system to track all the publications by active medical practitioners across the globe and provide an easy to track interface to leading pharmaceutical firms. The main source of the data is Pub-Med NCBI website, from which the data of publications are extracted. The interfacing API returns the data in XML format. The XML is then parsed and the data is sorted out intelligently to classify the physicians into numerous categories. The end clients are provided an attractive and interactive web interface through which they will have access to the entire data and can keep track of works of physicians they are interested in.

*Accomplishments*

* Designed and implemented database schema
* Developed stored procedures, cursors and functions  in Transact-SQL
* Created logical data models for database structure
* Performed performance tuning by creating indexes, optimization of queries, monitoring the query execution plan and database normalization.
* Created materialized views and indexes for performance.
* Developed Perl modules and packages
* Developed API to crawl through NCBI data base and fetch publication data.
* Used Perl DBI for database interaction and installed database drivers (DBD)
* Developed User Interface using Perl CGI, HTML, JavaScript, CSS
* Developed user front end form field authentication using Perl and JavaScript
* Employed Perl Template Toolkit to build and use Perl constructs with HTML
* Installed and maintained ISS, configured ISS for Active State Perl on Windows OS.
* Developed API in Perl to fetch and parse XML data using CPAN modules such XML::Parser and XML::Simple.
* Developed shell scripts for automation and running batch processes and load distribution.
* Implemented test framework in Perl for performing regression and unit testing.

**Education**

**Polytechnic Institute of New York University** Sep 2006 – Apr 2008

**Brooklyn, New York**

Master of Science in Computer Engineering